DEPARTMENT OF DEFENCE

DEFENCE SCIENCE & TECHNOLOGY ORGANISATION

DSTO

# Component-Based Engineering of Knowledge-Enabled Systems: Research Vision and Strategy

D.P.J. Goodburn, T.R. Pattison and R.J. Vernik

DSTO-GD-0271

# Component-Based Engineering of Knowledge-Enabled Systems: Research Vision and Strategy

*D.P.J. Goodburn, T.R Pattison and R.J. Vernik*

**Information Technology Division**
**Electronics and Surveillance Research Laboratory**

DSTO-GD-0271

## ABSTRACT

Australia's Strategic Policy [1] assigns the highest capability development priority to the achievement of the "knowledge edge" over our adversaries and identifies a single Command Support System that exhibits *flexibility, cost-effectiveness, robustness* and *adaptivity* as being an important enabler of the knowledge edge. This emphasis on the knowledge edge is reinforced in the Defence 2000 White Paper [2], which states that the knowledge edge "will be the foundation of our military capability over the coming decades". The Software Systems Engineering (SSE) group of ITD is currently conducting research into new software engineering principles and practices, collectively referred to as *component-based software engineering* (CBSE), which are expected to meet the requirements of cost-effectiveness and flexibility in the development of command support systems. In addition, *knowledge-based techniques,* and in particular *intelligent agents,* provide the opportunity to incorporate adaptivity and robustness into software systems through the use of machine-learning, automated reasoning and encapsulation of domain knowledge. In this paper we outline an extension of the Software Systems Engineering Group's research into CBSE to investigate ways in which intelligent agents can be used to exploit the combined advantages of component-based software engineering and knowledge-based techniques for the development of software for military applications.

**RELEASE LIMITATION**

*Approved for public release*

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

*Published by*

*DSTO Electronics and Surveillance Research Laboratory*
*PO Box 1500*
*Salisbury South Australia 5108 Australia*

*Telephone: (08) 8259 5555*
*Fax: (08) 8259 6567*

*AR-011-705*
*February 2001*

**APPROVED FOR PUBLIC RELEASE**

# Component-Based Engineering of Knowledge-Enabled Systems: Research Vision and Strategy

## Executive Summary

Australia's Strategic Policy assigns the highest capability development priority to the achievement of the "knowledge edge" over our adversaries and identifies a single Command Support System that exhibits *flexibility, cost-effectiveness, robustness* and *adaptivity* as being an important enabler of the knowledge edge.

The Software Systems Engineering (SSE) group of ITD is currently conducting research into new software engineering principles and practices, collectively referred to as component-based software engineering (CBSE), which are expected to meet the requirements of cost-effectiveness and flexibility in the development of command support systems. Knowledge-based techniques, on the other hand, provide an opportunity to incorporate adaptivity and robustness into software systems through the use of machine-learning, automated reasoning and encapsulation of domain knowledge. Unfortunately, few knowledge-based techniques have progressed beyond the laboratory. Recent research into software agents, in particular intelligent agents, has shown that knowledge-based techniques can be implemented in, or wrapped using, mainstream object-oriented programming languages, and hence made more amenable to conventional software engineering. Furthermore, since intelligent agents encapsulate their knowledge-based payloads, hiding the implementation details behind a narrow, well-defined interface, they are also well suited to incorporation within component software architectures as knowledge components.

In this paper we outline an extension of the Software Systems Engineering Group's research into CBSE to investigate ways in which intelligent agents can be used to exploit the combined advantages of component-based software engineering and knowledge-based techniques for the development of software for military applications. SSE Group's CBSE research is focused on the design and development of *InVision*, a component-based framework for system and information visualisation. This framework provides rich opportunities for the design and implementation of knowledge components to support interactions with the user, the information space, and other applications, including legacy systems.

Expected short-term outcomes from this research are:

- better positioning of DSTO to provide advice on the use of agents as knowledge components in military computing systems;
- the provision to Defence customers of component infrastructure for the flexible assembly and adaptive deployment of visualisation solutions;

- improved positioning of DSTO to provide advice on complete visualisation solutions, both through its own research program and through TTCP collaborations, in which the *InVision* infrastructure represents a solid Australian contribution.

A long-term goal of the *InVision* research program is to improve the flexibility and adaptivity of future ADF software systems in support of the more flexible command arrangements, including *ad hoc* coalitions, required by Australia's Strategic Policy. The component-based development of ADF software systems will lead to more cost-effective and robust military computing systems, while the *InVision* approach to software development and deployment will lead to the improved transfer of domain knowledge throughout their life cycle.

# Authors


## D.P.J. Goodburn

Information Technology Division


*Dr. Daniel Goodburn is employed as a Research Scientist in the Software Systems Engineering Group. His current research interests are software agents and systems visualisation. Previous work includes research into the use of expert systems to support mechanical engineering designers. He has a Bachelor of Science in the Faculty of Mathematical and Computer Science (Honours) from the University of Adelaide and a Ph.D. in Information Technology from the University of South Australia.*

---


## T.R. Pattison

Information Technology Division


*Tim Pattison received a B.Sc. (Ma) in Applied Mathematics and Computing in 1988, a B.E. (Hons) in Electrical and Electronic Engineering in 1989 and a Ph.D. in neural network modelling of early vision in 1993, all from the University of Adelaide, South Australia. As an undergraduate he held an Engineering Cadetship with the Electricity Trust of South Australia, and later an Undergraduate Cadetship with the Advanced Engineering Laboratory (AEL) of the Defence Science and Technology Organisation (DSTO). His postgraduate research was supported by a Postgraduate Fellowship from the Communications Division of DSTO, and in part by the Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP), Pooraka, South Australia.*

*Between 1993 and 1996, Dr Pattison worked as a Research Scientist in the Satellite Networks Discipline of Communications Division, primarily on the signal processing and geometrical aspects of dual-satellite geolocation. He then moved to C3I Networks Discipline, working firstly on issues concerning graceful degradation and the optimisation of delivered value in communications networks, and later as leader of Organisational Networks and Processes Section. In 1999 he transferred to Software Systems Engineering Group of Information Technology Division, where he now leads the task entitled "Assembly and Deployment of Defence Visualisation Solutions", concerned with the research and development of information visualisation systems and their application to Recognised Enterprise Pictures.*

---

## Authors (Cont)


### R. Vernik

Information Technology Division


*Dr. Rudi Vernik is employed as a Principal Research Scientist and is Head of Software Systems Engineering Group. His research interests focus on the definition, development, and application of new systems and software engineering approaches to support Defence in its development of capabilities that will facilitate knowledge and information-based warfare. His group is currently undertaking research in the areas of systems visualisation, component-based software engineering, systems characterisation and modelling, systems dynamics, and evolutionary capability development processes. Rudi has a Ph.D. in Computer and Information Science (Software Engineering) from the University of South Australia. He also has a Bachelor of Electronics Engineering (with Distinction) and a Diploma of Communications Engineering from the Royal Melbourne Institute of Technology.*

# Contents

# 1. Introduction

Australia's Strategic Policy [1] assigns the highest capability development priority to the achievement of the "knowledge edge" over our adversaries. To see further and more clearly through the fog of war, the ADO requires not only better capabilities in the acquisition and dissemination of information, but also better exploitation of available information sources.

Knowledge-based techniques have already demonstrated their potential for facilitating this exploitation. Techniques such as machine learning, pattern recognition and data fusion have been used successfully in intelligence and surveillance applications to distil knowledge from large volumes of data. Decision aids which encapsulate domain knowledge, intelligent user agents which learn user preferences, and even conventional software which automates repetitive tasks and organisational processes, all serve to "speed up and improve command decision-making" [1] by freeing up the decision-maker to concentrate on knowledge-intensive activities which are less easily automated.

It is hardly surprising, therefore, that Australia's Strategic Policy identifies a single "robust and cost-effective" command support system (CSS) as an important enabler of the knowledge edge. This CSS needs to support "adaptability and flexibility" in command arrangements. Software Systems Engineering (SSE) group of ITD is currently conducting research into new software engineering principles and practices, collectively referred to as *component-based software engineering* (CBSE), which are expected to help meet two of these requirements. Cost-effectiveness will be achieved through the extensive reuse of commercial off-the-shelf (COTS) and government off-the-shelf (GOTS) software components, while CSS flexibility will be improved through support for the rapid assembly of light-weight applications which are tailored to the application domain.

Rapid application assembly will be essential for the future support of short-notice operations, a recent example of which was the deployment in East Timor. The robustness of Australia's CSS once deployed, however, will depend on software which is more adaptive to changes in the information environment, organisational processes and user requirements. Such changes are especially likely to occur in operations, such as the East Timor deployment, which require interoperability within ad hoc coalitions. Many knowledge-based techniques are inherently adaptive, and are therefore obvious candidates for achieving the required levels of software adaptivity. However, of those techniques which have proven to be effective in experimental prototypes, few have progressed beyond the laboratory. This is due not only to the cost of the specialist programming skills required and the difficulty of reusing application-specific code, but also to the absence or immaturity of applicable software engineering principles and practice [3], [4].

Recent work in software agents, and in particular intelligent agents, has shown that knowledge-based techniques can be implemented in, or wrapped using, mainstream

object-oriented programming languages, and hence made more amenable to conventional software engineering. Since intelligent agents encapsulate their knowledge-based payloads, hiding the implementation details behind a narrow, well-defined interface, they are also well suited to incorporation within component software architectures as knowledge components. This position paper outlines the extension of Software Systems Engineering Group's research into CBSE to investigate ways in which intelligent agents can be used to exploit the combined advantages of component-based software engineering and knowledge-based techniques for the development of software for military applications. SSE Group's CBSE research is focused on the design and development of *InVision*, a component-based framework for system and information visualisation. This framework provides rich opportunities for the design and implementation of knowledge components to support interactions with the user, the information space, and other applications, including legacy software.

The component-based approach is distinguished from agent-based software engineering [3] in that the principal programming abstraction is the software component rather than the agent. This choice will facilitate the use of currently available COTS software components for functions which do not require the advanced behavioural properties characteristic of agents.

# 2. Agents as Knowledge Components of Defence Systems

## 2.1 What are agents?

Unlike traditional software objects, software agents are goal-directed in that they [5]:

- pursue their goals autonomously and proactively;
- use social abilities such as communication, coordination and negotiation to enlist other agents in the pursuit of their goals;
- decide whether and how to assist others based on the consistency between the requested actions and their own goals; and
- adjust their goals in response to changes in their environment.

Multi-agent systems are inherently multi-threaded, since each agent must be continuously aware of its environment, and able to sense and act on that environment in the pursuit of its goals. It will be argued shortly that this property is both a strength and a weakness. Several agent frameworks also provide for agent mobility – the ability to suspend execution, relocate to a different computing platform, and resume execution.

## 2.2 Why use agents?

Military computing systems are open, distributed and complex. In order to illustrate the meaning of these terms, their implications for the design and implementation of command support systems, and the potential benefits of software agents, we will use the example of the design of an organisation.

An organisation, like a military computing system, is an example of an open, distributed and complex system. An open system is one whose components are not all known or adequately characterised in advance, and may change with time. A complex system consists of a large number of heterogeneous elements – including in this case the people, information technology and other infrastructure. Many of these elements are in turn complex systems, although complexity can also exist in systems with simple elements. In a distributed system, responsibility, authority, knowledge, expertise and resources are widely distributed across the elements of the system. This is clearly true of all but the smallest of organisations.

The conventional approach to organisational design is to characterise the external environment, define the mission and goals of the organisation to fulfil demands from, or exploit opportunities in, that environment, and design the organisational structure based on a functional decomposition of these goals into sub-goals. Although interaction between peers is invariably required, it is only rarely defined in any detail in advance. Once deployed in a position, an individual will establish interactions which assist in the fulfilment of his or her assigned functions. Since the goals of the organisation will evolve over time in response to changes in the environment, both the entities with which it interacts and the nature of those interactions will change in ways which cannot be predicted in advance. In order to adapt rapidly to these changes, a modern organisation typically exhibits bottom-up organisation of personnel into teams, with team members aligning their individual goals with those of the team. The more enduring changes may also trigger a top-down adaptation of the organisation's goals and sub-goals, the formal roles and responsibilities of its personnel, and their interactions.

The organisational flexibility and adaptability advocated in Australia's Strategic Policy [1] will be increasingly important if the current range of Military Response Options is to be sustained while the Defence budget shrinks in real terms. In order to adequately support the future ADF, our command support system(s) will need to be capable of corresponding levels of flexibility and adaptivity. Although not pre-requisites for agenthood, the individual adaptivity, flexible communication and collective self-organisation required by modern organisations and computing systems alike are common properties in software agents. Agents are therefore likely to be important components in future CSSs.

Indeed, agent-oriented programming has been proposed as a natural design and implementation paradigm for such open, distributed and complex systems [6]. It is our contention, however, that just as not all of the assets held by an organisation require the autonomy and adaptivity characteristic of its personnel, so not all of the

components of a CSS will warrant the programming and performance overheads of software agents. Furthermore, the possibilities of undesirable emergent behaviour and execution dead-locks [6] indicate that agents will remain unsuitable for certain trusted applications. We therefore propose the use of agents only where agent functionality is specifically indicated.

## 2.3 Component-based software engineering

The SSE group is currently conducting research into component-based software engineering (CBSE), an emerging discipline which focuses on the rapid assembly of software solutions from reusable software assets. Implemented using component technologies such as JavaBeans, CORBA, COM and ActiveX, these assets include in-house, off-the-shelf and third party software components, and component frameworks which provide the infrastructure for integrating components. CBSE has the potential to reduce the cost of software development through component re-use, and through the commoditisation of components conforming to (often de facto) standard frameworks. A focus on component assembly rather than code development will accelerate the application development cycle, leading to the rapid assembly of light-weight, tailored applications, rather than reliance on off-the-shelf, monolithic, one-size-fits-all applications. The encapsulation, and hence localisation, of well-defined functionality within components facilitates software maintenance, while hiding the implementation of this functionality behind an unchanging interface facilitates software upgrades and the military customisation of COTS software.

Whilst the rapid assembly and re-assembly of tailored applications will support the goal of flexibility in military computing systems, component-based software engineering does not by itself lead to software which is adaptive to changes in the deployment environment. Although a shorter software development cycle would free up developers to support more frequent, step-wise adaptations of the software, the prohibitive cost of on-tap software engineers and the down-time associated with software upgrades suggests that automated adaptation is required. Also, since some changes – such as accommodating the evolving preferences of individual users – happen on a time scale and numerical scale which could not be easily accommodated by manual changes, continuous, automated adaptation is the only real option.

Knowledge-based techniques such as machine learning offer a considerable degree of adaptivity. However, many have not progressed beyond implementation as monolithic laboratory prototypes. The cost of specialist programming skills, the difficulty of re-using monolithic, application-specific code and the lack of applicable software engineering principles and practices have all conspired to limit the transition of knowledge-based techniques from the laboratory into engineered products. The use of mainstream, object-oriented languages such as C++ or Java (except where non-procedural programming is strongly indicated), and the development of component-based frameworks for knowledge-based techniques, would help address these problems by reducing development costs and facilitating code re-use. We propose the use and development of component-based software engineering principles and

practice to facilitate the incorporation of knowledge-based techniques into knowledge-enabled applications.

## 2.4 Agents as components

Intelligent software agents are an obvious technology for this synthesis, since they are both knowledge-based and – like software components – largely self-contained. The encapsulation of knowledge-based functionality within the agent's "payload" localises the requirement for specialist programming skills. The ability of agents to collectively self-organise provides additional scope for software adaptivity, even in cases where the individual agents would not be considered particularly "intelligent". The development of a new software engineering discipline entitled "Agent-Based Software Engineering", in which the principal programming abstraction is the software agent, has been proposed [6] to exploit this potential for the implementation of open, complex, distributed and adaptive systems. However, for the simple functions which do not require the advanced behavioural properties of agents, the development overhead involved in designing and debugging the multi-threaded code required by agent implementations would be an unnecessary imposition. Furthermore, the potential emergence of undesirable behaviour makes agent-based implementation undesirable in trusted computing applications. Component-based software engineering, on the other hand, uses the software component as its principal programming abstraction. We contend that CBSE, when adapted to incorporate agents as components, offers a more promising approach to the development of tomorrow's knowledge-enabled applications.

## 2.5 Research issues

The component-based implementation of military computing systems which incorporate software agents to assist with domain embedding raises many important research issues. SSE group has chosen to address those which fall into the following four categories:

- **Agent integration into CBSE**: what modifications to, or constraints on, software agents and agent frameworks, and to CBSE principles and practices, are required for integration of the former into the latter?
- **Hybrid software design**: how do we trade off the conflicting requirements of limiting the use of software agents and achieving the required adaptivity?
- **Hybrid software implementation**: how do we identify the appropriate agent technologies for the stated requirements?
- **Knowledge transfer**: how do we represent and initialise agent knowledge of the domain context, and retrieve and store it for re-use when application re-assembly becomes necessary?

Research into these topics will complement the existing DSTO agent research summarised in Appendix A.

# 3. Research Strategy – *InVision*

These questions are being addressed by SSE group under the VIDECS task (JNT 97/127) in the research and development context of *InVision,* a component-based approach to the development and deployment of computer-based solutions for system and information visualisation. The field of visualisation is appropriate in that not only does it have widespread application within Defence, but it also offers rich opportunities for the provision of supporting functions, many of which would benefit from agent-based implementation.

## 3.1 *InVision* approach

The *InVision* approach to software development and deployment is depicted in Figure 3-1. The process begins with the capture and modelling of the domain context into which the visualisation solution will be deployed. The term "domain context" here encompasses the work processes and individual user preferences which are to be supported, the information space from which the information to be visualised must be collected, and the available IT tools and services with which the deployed visualisation solution must interoperate. Based on a firm understanding of the domain context, a visualisation tool is then assembled from a set of component assets. The relevant domain knowledge is incorporated through explicit representation in models of the domain context, and in the agents' internal representations of the environment. Domain knowledge is also represented implicitly in the choice of components to be assembled, the organisation of user workspaces, and in the targeting of adaptive functionality towards those aspects of the environment which are expected to change most significantly. In-service operation is characterised by (largely) agent-mediated interactions with the domain context, as represented by the green "halo" around the *InVision* solution in Figure 3-1. The appropriate degree of adaptivity in these interactions lessens the frequency with which the whole process needs to be repeated. In the event that the inevitable limits to this adaptivity are exceeded, existing domain knowledge distilled by use monitoring and agent learning functions would be fed back to the domain modelling process in preparation for reassembly and deployment of the solution in line with changing requirements and conditions.

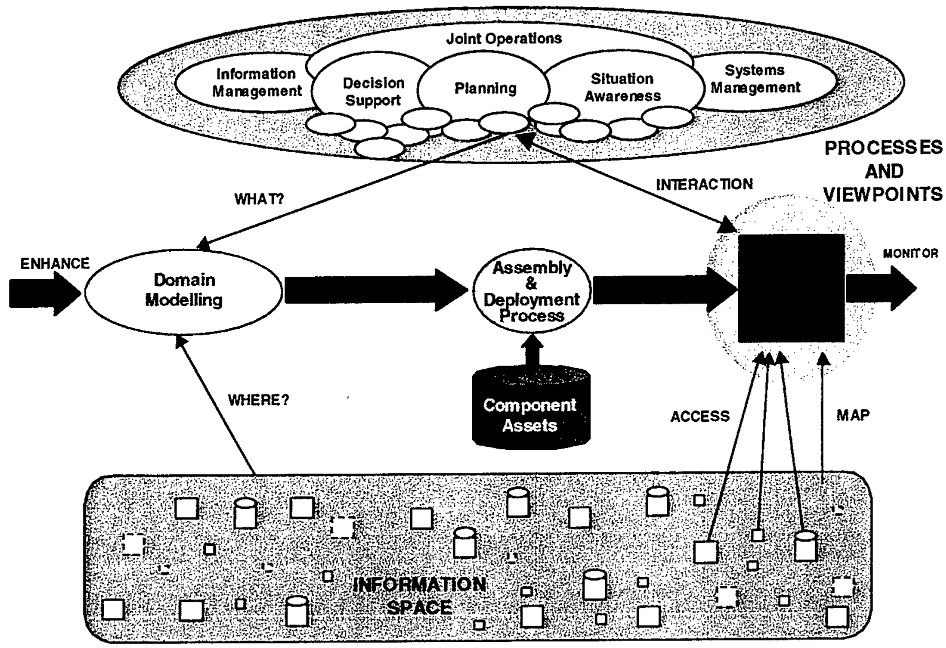


*Figure 3-1: The InVision approach to the assembly and deployment of knowledge-enabled systems within their domain context.*

We note in passing that although the *InVision* approach to software development and deployment is currently being developed and applied in the field of visualisation, it is expected to also apply to a broad range of other applications in the future command support systems of the ADF.

## 3.2 *InVision* agents

SSE group has proposed a component-based software architecture, and is currently defining and developing the constituent frameworks, which will enable the flexible assembly and deployment of visualisation solutions. Shown in Figure 3-2 in conceptual form, the architecture calls for the use of agents to mediate the Use, Provision and Interoperation processes, which involve interactions with the users and their work processes, the information space, and the available tools and services respectively. For convenience, agent roles are currently categorised into corresponding Use, Provision and Interoperation categories, although it is anticipated that some may need to span the boundaries between these classifications. Subdivision of these categories is also possible; for example, Provision agents include Exploration agents, which are responsible for mapping the information space, and Collection agents, responsible for collecting the required information from that space.

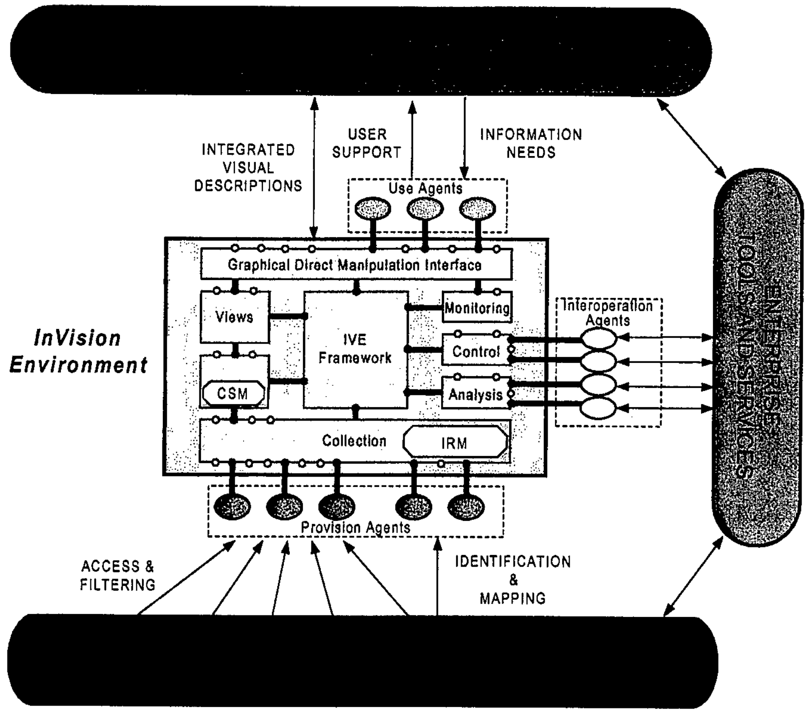


*Figure 3-2: **Conceptual representation of the InVision architecture.***

Implementation of these agent roles will provide rich opportunities for the investigation of the full range of agent capabilities and frameworks, including agent mobility, multi-agent systems and collective self-organisation. For example, in order to expedite exploration of the information space, the current mobile, multi-agent implementation [7] of the Exploration role allows agents to explore in parallel the file systems on different host computers. The amount of "intelligence" required by an individual agent will also vary from the relatively un-intelligent, such as inferring file type from file name extension, magic number and other cues, through to the highly intelligent, such as inferring and accommodating the user's visualisation goals as they evolve during information exploration.

## 3.3 Agent research

In this section, we elaborate on the research issues identified in Section 2.5 in the context of *InVision* research.

### 3.3.1 Agent integration into CBSE

A number of agent frameworks, specialising in (overlapping) subsets of agent properties, are now available. Examples include Aglets [8] and Concordia [9] for agent mobility, and JACK [10] for beliefs, desires and intentions (BDI). Because of the large variety of roles to be played by agents within the *InVision* framework, it is unlikely that

any one agent framework will support the implementation of all possible *InVision* agents. Therefore the *InVision* framework will need to accommodate *various* agent frameworks.

*InVision* research will investigate appropriate interfacing, adaptation and wrapping strategies to allow existing agents and agent frameworks to be packaged as components, and any modifications to agent implementation or framework design practices which will facilitate their integration into component-based software as first-class components.

### 3.3.2 Hybrid software design

It is important to ensure that greater adaptivity in ADF command support systems is not achieved at the expense of reduced robustness. Because of the potential disadvantages of agent-based implementation identified in Section 2.4, it is important to confine agent functionality to specific roles which are identified as needing "intelligence" or adaptivity beyond the capabilities of non-agent components. This work will begin by identifying and classifying *InVision* functions which require, or will benefit substantially from, agent-based implementation. The resultant taxonomy will need to evolve as new functions are conceived and added. In addition, strategies for quarantining both emergent behaviour and problems with deadlock or livelock in agent components from non-agent components, which implement more trusted functionality, will be investigated. Those conventional components which interact with agents will need to be capable of recovering from the failure of an agent to complete its task, and to invoke fallback strategies, which may use non-agent implementations to achieve a possibly less ambitious result. Interfaces between agent and non-agent components will also need to limit the scope of what an agent can ask a conventional component to do, thus reducing the ability of renegade agents to interfere with the core functionality of the application.

### 3.3.3 Hybrid software implementation

An *InVision* function which has been identified as requiring agent implementation must then be mapped onto the appropriate agent technology. The choice of agent framework should be made primarily on the basis of the match between the agent properties – such as mobility, intelligence or self-organisation – which are supported by each framework, and those required for implementation of the function. The preparation and maintenance of a list of available agent frameworks, classified on the basis of the agent properties they support, will assist in making this choice. However identification of the required agent properties may not be straightforward, since, for example, the required adaptivity could potentially be achieved through either one rather intelligent agent or the self-organisation of a number of less intelligent agents. These two options would call for quite different agent frameworks. In other cases where an *InVision* function has a number of disparate requirements which cannot be satisfied by any single agent framework, it may prove necessary to split the

implementation between several framework and provide mechanisms for interaction between different agent frameworks.

The experience required to decide between these different design alternatives can only come from practice in agent-based implementation of *InVision* functions. Initial efforts will concentrate on the implementation of the Collection framework. These efforts will also aim to identify any shortfalls in current agent technology and, where appropriate, contribute to overcoming these shortfalls.

### 3.3.4 Knowledge transfer

The *InVision* approach described in Section 3.1 involves the distillation of knowledge about the domain context, both prior to application assembly and during its in-service use. The knowledge obtained during the domain characterisation phase needs to be incorporated in the agents and their interactions during application assembly. Similarly, the knowledge distilled by these agents during in-service operation needs to be available for reuse in subsequent iterations of the domain characterisation and application assembly processes.

Some of this knowledge is stored in shared models, such as the Composite Systems Model, on which the views are based. Much of it will be encapsulated by agents in their internal models of the environment, and represented in their learnt patterns of interaction. In addition to the issues arising from the need for knowledge transfer, SSE group will develop strategies for the division of knowledge representation responsibilities in an *InVision* assembly which takes into account the relative accessibility of knowledge stored by these three mechanisms.

## 3.4 Strategy

The *InVision* approach and architecture, developed under the VIDECS task, provides an excellent platform for motivating, testing, integrating and transitioning the results of agent research. Ongoing *InVision* research will draw on expertise within SSE group, other DSTO groups, and academia to investigate the specific research issues identified in the previous section.

Building on its extensive experience in software engineering, SSE group is currently undertaking research into component-based software engineering. Through this work, we have gained experience in the design of component system architectures, the development of components and component frameworks using JavaBeans, and the integration of third-party components. In addition, a number of Defence domain case studies have given us a good understanding of the application of visualisation in a broad range of Defence settings, including an appreciation of the deployment requirements which are likely to be served by agent implementation. These skills, coupled with research into agent applications in *InVision* [11] and recent experience with the use of the Aglets agent frameworks mean that we would be well-placed to perform much of the identified research in-house. However, the inevitable limitations on the number of allocated staff mean that we will need to contract out some of the

work, and to be on the lookout for opportunities such as the specification and supervision of vacation and Honours student projects. We will also seek research collaborations in selected areas, such as the application of knowledge-based techniques, in which we have insufficient expertise.

Appendix A presents a survey of existing DSTO research into software agents and their applications. SSE group will exploit the results of this research, and the expertise of DSTO researchers, wherever relevant. In particular, our agent research is likely to benefit both from the use in the future *InVision* analysis framework of the information fusion architectures listed in Appendix A.2, and from extensive DSTO experience with BDI agents. Our work will complement, rather than compete with, these other DSTO research threads.

The *InVision* research program offers broad scope for collaborative research into software agents and their integration into component architectures. SSE group has several existing collaborations which are either already contributing to our agent research program, or have the potential to do so. Our existing relationship with the Flinders University has resulted in the development of a mobile agent system for exploration of a distributed information space (see Appendix A.4). This work provides a starting point for research and development of the *InVision* Collection Framework shown in Figure 3-2. We are also collaborating with members of the Distributed Systems Technology Centre (DSTC) on the ENE project, with a view to the possibility of using the Elvin event notification service as an agent communication architecture. And finally, we are keeping a watching brief on the use of agents in the DSTC's Ecology project.

In addition to these existing collaborations, SSE group will seek to contract out, to the relevant centres of expertise, the implementation of agent "smarts" using knowledge-based techniques such as machine learning and expert systems. This and any other research which has the potential to contribute to the *InVision* research program, and which would benefit from the broad research vision, component infrastructure and ready-made client base of *InVision*, will be incorporated via targeted collaborations using open software principles. It is anticipated that making the *InVision* component assets freely available in this way will encourage the development of an *InVision* developer community, and thereby improve and extend the *InVision* component base as developers provide feedback on existing infrastructure and contribute their own *InVision* components and frameworks.

## 3.5 Transition of research into Defence

We have argued that software assembly from a component infrastructure offers greater flexibility for future ADF software systems. The incorporation of agents as knowledge components, and their use to support software deployment, promises the additional benefit of greater in-service adaptivity. The *InVision* component infrastructure provides the vehicle to both attract third-party contributions, and to transition these and other results of *InVision* research into Defence use. The deployment of an experimental

*InVision* server for Defence use will be investigated. One or more demonstration assemblies will be developed for use on the server, in order to both elicit requirements for, and prioritise research and development of, the component infrastructure. In cases where a Defence customer requires the development of a tailored visualisation application based on *InVision*, SSE group will advise on the resources which will need to be committed, and provide technical oversight of the development effort.

# 4. Defence Outcomes

A long-term goal of the *InVision* research program is to improve the flexibility and adaptivity of future ADF software systems in support of the more flexible command arrangements, including ad hoc coalitions, required by Australia's Strategic Policy [1]. The component-based development of knowledge-enabled systems will lead to more cost-effective and robust military computing systems, while the *InVision* approach to software development and deployment will lead to the improved transfer of domain knowledge throughout their life cycle.

In the shorter term, DSTO will be better positioned to provide advice on the use of agents as knowledge components in military computing systems. Additional outcomes arising from the choice of information and system visualisation as the application domain for this research include: the provision to Defence customers of component infrastructure for the flexible assembly and adaptive deployment of visualisation solutions; and the improved positioning of DSTO to provide advice on complete visualisation solutions, both through its own research program and through TTCP collaborations, in which the *InVision* infrastructure represents a solid Australian contribution.

# 5. References

[1] Australian Department of Defence, *Australia's Strategic Policy*, Directorate of Publishing and Visual Communications, 1997.

[2] Australian Department of Defence, Defence 2000: Our Future Defence Force, Commonwealth of Australia, 2000, p. 94.

[3] M. Wooldridge, "Agent-based software engineering," *IEE Proc. Softw. Eng*, vol. 144, pp. 26-37, 1997.

[4] M. D. Wilson, "Why aren't SE techniques being used by KBS systems developers already?," , vol. 2000, 1989.

[5] M. Wooldridge, "Intelligent Agents," in *Multiagent systems: a modern approach to distributed intelligence*, G. Weiss, Ed. Cambridge, Massachusetts: The MIT Press, 1999, pp. 27-77.

[6] N. R. Jennings and M. Wooldridge, "Applying Agent Technology," *Applied Artificial Intelligence: An International Journal*, vol. 9, pp. 351-361., 1995.

[7] C. S. Dickson, "Automatic Classification of Information in a Distributed Environment," in *Computer Science*. Adelaide: Flinders University, 1999, pp. 55.

[8] aglets.org, "www.aglets.org - a resource for the Java Aglet community", [Online accessed 12 December, 2000] URL: http://www.aglets.org

[9] Mitsubishi Electric Information Technology Center America, "Concordia - Java Mobile Agent Technology", [Online accessed 12 December, 2000] URL: http://www.meitca.com/HSL/Projects/Concordia/

[10] Agent Oriented Software, "Agent Oriented Software - JACK Intelligent Agents", [Online accessed 12 December, 2000] URL: http://agent-software.com.au/jack.html

[11] D. P. J. Goodburn, R. J. Vernik, M. P. Phillips, and J. J. Sabine, "Integrated Visualisation and Description of Complex Systems," DSTO, Research Report DSTO-RR-0154, 1999.

[12] A. Lucas and S. Goss, "The Potential for Intelligent Software Agents in Defence Simulation," presented at Information, Decision and Control 99, 1999 Information, Decision and Control : Data and Information Fusion Symposium, Signal Processing and Communications Symposium and Decision and Control Symposium, Adelaide, Australia, 1998.

[13] P. Clark, H. Pongracic, M. Cross, and J. Morton, "Incorporating Intelligent CGF in a Human-in-the-Loop Simulation," presented at Simulation and Technology Conference (SimTecT 99), Melbourne, Australia, 1999.

[14] R. Evertsz and R. Ronnquist, "Agent-Oriented Control of Battlefield Simulations," presented at SimTecT98, Adelaide, Australia, 1998.

[15] M. Anderson, D. Engelhardt, C. Fiddyment, D. Marriott, C. North, and D. Rajaratnam, "Shapes Vector: An Overview," Defence Science & Technology Organisation, Technical Report 1998.

[16] D. Lambert, "Advisers with Attitude for Situation Awareness," presented at Data and Signal Processing II, Chicago, USA, 1999.

# Appendix A: DSTO Research

The DSTO has a number of research threads involving the use of agents. These include

- Defence simulation
- Information fusion
- Situation awareness and mission planning
- Information discovery

## A.1. Defence simulation

The simulation of Defence operations provides a cost-effective means of training personnel, investigating human performance factors, developing tactics and doctrine, and informing capability development decisions. Friendly and opposing forces can be modelled at the appropriate level(s) of aggregation and detail, and their capabilities varied in a range of scenarios to analyse the effect on engagement outcomes. Agent-based models allow both the direct variation of these capabilities at the lowest level of aggregation represented in the simulation – which may or may not be individual warfighters or platforms – and the flexible composition of forces from these individual agents.

The use of agents in Defence simulation is a well-established research theme in the Air and Land Operations Divisions (AOD and LOD respectively) of DSTO. The AOD has developed the SWARMM air mission simulation system and is currently developing the BattleModel as the basis for future capability modelling in the AEW&C Support Facility for an Australian AEW&C aircraft (Project Wedgetail) [12]. The BattleModel provides a "plug-and-play" architecture allowing a number of different agents to be selected depending on a given scenario and incorporated into the model. Agents, developed using dMARS (Distributed Multi-Agent Reasoning System), are used in the BattleModel as computer generated forces, representing either friendly or enemy pilots, and provide a natural way of simulating the behaviour of either individuals or teams of pilots [13].

Land Operations Division has developed the CAEN/dMARS demonstrator to illustrate the potential of agents in the simulation of land operations. Simulation gives commanders the ability to evaluate in advance the various options available to them through multiple runs of a simulation in a controlled environment. CAEN (Closed Action ENvironment) is used for tactical simulations at the company level. Agents are used to model aggregated entities, such as platoons and companies, and the reasoning processes which drive the associated command and control structure [14].

Both the BattleModel and CAEN demonstrators are examples of multi-agent systems – systems in which each component is an agent – and provide strong platforms for research into agent issues such as cooperation, learning, negotiation, intention recognition, human and team modelling.

## A.2. Information fusion

Information fusion involves distilling large amounts of information into knowledge through the removal of redundancy, deconfliction, and conceptual abstraction. For large or poorly constrained problem domains, the ontology – the specification of a vocabulary to be used in reasoning about the domain to which the information relates -- is often highly complex, consisting of words at different levels of abstraction and with many inconsistencies and ambiguities. One way to overcome this problem is to break the vocabulary into smaller, consistent, manageable chunks at different layers of abstraction, and assign to each chunk a specialised agent. These agents then cooperate to generate higher-level knowledge from the supplied information.

### A.2.1 Network security

The Advanced Computer Capabilities Branch is conducting research into the use of agents to support the fusion of low-level data from a monitored computer network to infer higher-level knowledge about the security of the network [15].

A hierarchical agent architecture is used to manage the complexity of the domain ontology. The ontology is divided into several abstraction layer ontologies, with each ontology layer having a higher level of abstraction than the layer immediately below it. The ontology used within each layer is internally consistent. Agents are then assigned to particular layers depending on the knowledge abstractions they use. Agent communication is restricted by allowing agents to communicate only with agents at different levels. This reduces inconsistencies and ambiguities in the layer ontologies. The result of this architecture is a hierarchical framework of agents. Agents at the lowest layer process large amounts of raw data from a computer network with results being passed up to the next higher layer. Each successive layer of agents produces knowledge at a higher level of abstraction. Thus the large amount of raw data is reduced to a much smaller, high-level picture of the computer network that is easily interpreted by human users.

### A.2.2 Intelligence gathering

A large amount of intelligence information can be obtained from open-source documents such as news articles and reports from government agencies. Maintaining up-to-date knowledge in domains of interest to the Defence user requires sifting through thousands of lines of text daily.

In ITD, the Information Management and Fusion (IMF) group is conducting research into fact extractors – simple agents which work together to extract facts from formatted or free text documents. Knowledge required for natural language processing of the text is segregated into narrow domains of expertise, each of which is assigned to a corresponding fact extractor. A fact extractor may either access the text directly, or observe the outputs of several others. In the area of maritime surveillance, for instance, there may be fact extractors for ships, cargo, dates and locations. A higher-level fact

extractor might employ these fact extractors to discover the fact, and associated details, of a ship arrival at port.

### A.2.3 Surveillance

The Surveillance Systems Assessment group within the Surveillance Systems Division is currently undertaking research in the use of agent negotiation as a means of scheduling the tasks of multifunction radar. A ship self-defence system is being modelled which consists of two platforms, one sea- and one air-based. The platforms, each having a 2D multifunction radar and radar controller, are connected by a data link. The aim of the research is to look at how link constraints such as bandwidth and latency effect this system, and how agent knowledge can be used to reduce the network traffic. The ATTITUDE multiagent reasoning system [16], developed at DSTO, is used to model the radars and controllers. ATTITUDE aims to lift the level of interaction abstraction between people and computers above that of high-level programming languages and graphical user interfaces to a level that is closer to the way people interact with other people. Knowledge inside ATTITUDE agents is represented in terms of propositional attitudes e.g. Fred believes that (blue sky), and ATTITUDE agents are programmed using propositional attitude instructions e.g. Fred believe (blue sky). In this way, explaining the behaviour of ATTITUDE agents is similar to explaining the behaviour of other people by attributing a mental state to them. Likewise, ATTITUDE agents are programmed through instructions that are similar to those that might be imparted to a person.

## A.3. Situation awareness and mission planning

During face-to-face interactions, a person conveys information using not only vocal, but also visual cues such as gestures and facial expressions. Animated agents try to capture some of these aspects of human communication and have been investigated as a means of presenting information to users in a way that is natural and easy to understand.

As part of its research into the Future Operations Centre Analysis Laboratory (FOCAL), Human Systems Integration (HSI) group of ITD will investigate the use of animated agents within a virtual reality world. These agents will act as a panel of collaborating advisers to users of a future operations centre, providing users with situation reports and walking them through relevant information. The FOCAL research will investigate aspects of agent-agent and agent-person interactions within virtual environments. Agents developed as part of FOCAL will be based upon the ATTITUDE multiagent reasoning system described above.

## A.4. Information discovery

In order to efficiently locate information stored on a corporate computer network, up-to-date meta-information about file content must be maintained. Whilst this is often done manually, in the form of hand-crafted HTML files, for the small fraction of files

which are made available via the intranet or exported to the Internet, the majority of files on the typical network remains un-categorised. The first step towards the efficient location of information is the automated recognition of file type, so that the appropriate tools can then be employed to categorise the file content without time-consuming human intervention.

The SSE group has conducted research into the use of mobile agents to find and classify the information sources available on an intranet [7]. Each agent traverses the intranet, categorising files locally on each computer before either returning to the original host with the classifications it has made or sending its results remotely. Multiple agents may be spawned to work in parallel, reducing the time required to traverse the entire network and analyse each computer. This research provides a basis for the Exploration Agents discussed as part of the *InVision* system in Section 3.

## A.5. Summary

Viewed as a whole, DSTO research into agents provides good coverage of many aspects of agent research, including agent architectures and interactions, and their application to a variety of domains. Higher level issues concerning agent system architectures and agent frameworks have also been touched upon by this research. These higher level issues are of particular interest to the SSE group.

DISTRIBUTION LIST

**Component-Based Engineering of Knowledge-Enabled Systems**

*D.P.J. Goodburn, T.R. Pattison and R.J. Vernik*

| | Number of Copies |
|---|---|
| **AUSTRALIA** | |
| **DEFENCE ORGANISATION** | |
| **Task sponsor:** | |
| DGC4 | 1 |
| Martin Faulkner, DDBD | 1 |
| **S&T Program** | |
| Chief Defence Scientist ) | |
| FAS Science Policy ) | 1 shared copy |
| AS Science Corporate Management ) | |
| Director General Science Policy Development | 1 |
| Counsellor, Defence Science, London | Doc Control Sheet |
| Counsellor, Defence Science, Washington | Doc Control Sheet |
| Scientific Adviser to MRDC Thailand | Doc Control Sheet |
| Scientific Adviser - Policy and Command | 1 |
| Navy Scientific Adviser | Doc Control Sheet and distribution list |
| Scientific Adviser - Army | Doc Control Sheet and distribution list |
| Air Force Scientific Adviser | 1 |
| Director Trials | 1 |
| **Aeronautical & Maritime Research Laboratory** | |
| Director | 1 |
| **Electronics and Surveillance Research Laboratory** | |
| Director | Doc Control Sheet and distribution list |
| Chief Information Technology Division | 1 |
| Research Leader Command & Control and Intelligence Systems | 1 |
| Research Leader Military Information Enterprise | 1 |
| Research Leader Advanced Computer Capabilities | Doc Control Sheet |
| Research Leader Joint Systems | 1 |
| Head, Information Warfare Studies Group | Doc Control Sheet |
| Head, Trusted Computer Systems Group | Doc Control Sheet |
| Head, Systems Simulation and Assessment Group | Doc Control Sheet |
| Head, C3I Operational Analysis Group | Doc Control Sheet |
| Head, Information Exploitation Group | Doc Control Sheet |
| Head, Intelligence Group | Doc Control Sheet |

| | |
|---|---|
| Head, Human Systems Integration Group | Doc Control Sheet |
| Head, C2 Australian Theatre | 1 |
| Head, Distributed Systems Group | Doc Control Sheet |
| Head C3I Systems Concepts Group | 1 |
| Head, Military Systems Synthesis Group | Doc Control Sheet |
| Head, Systems of Systems Group | Doc Control Sheet |
| Head, Advanced Network Integrity Group | Doc Control Sheet |
| Head, Enterprise Visualisation, Instrumentation and Synchronisation Group, ITD (Dr Rudi Vernik) | 1 |
| Dr Tim Pattison EVIS Group, ITD | 1 |
| Dr Daniel Goodburn, EVIS Group, ITD | 1 |
| ITD Publications & Publicity Officer, ITD /Executive Officer, ITD | 1 shared copy |

**DSTO Library and Archives**

| | |
|---|---|
| Library Fishermans Bend | Doc Control Sheet |
| Library Maribyrnong | Doc Control Sheet |
| Library Salisbury | 1 |
| Australian Archives | 1 |
| Library, MOD, Pyrmont | Doc Control Sheet |
| US Defense Technical Information Center, | 2 |
| UK Defence Research Information Centre, | 2 |
| Canada Defence Scientific Information Service, | 1 |
| NZ Defence Information Centre, | 1 |
| National Library of Australia, | 1 |

**Capability Systems Staff**

| | |
|---|---|
| Director General Maritime Development | Doc Control Sheet |
| Director General Land Development | Doc Control Sheet |
| Director General Aerospace Development | Doc Control Sheet |

**Knowledge Staff**

| | |
|---|---|
| Director General Command, Control, Communications and Computers (DGC4) | Doc Control Sheet |
| Director General Intelligence, Surveillance, Reconnaissance and Electronic Warfare (DGISREW | Doc Control Sheet |
| Director General Defence Knowledge Improvement Team (DGDKNIT) | Doc Control Sheet |

**Army**

| | |
|---|---|
| Stuart Schnaars, ABCA Standardisation Officer, Tobruck Barracks, Puckapunyal VIC 3662 | 4 |
| SO (Science), Deployable Joint Force Headquarters (DJFHQ) (L), MILPO, Gallipoli Barracks, Enoggera, QLD 4052 | Doc Control Sheet |

**Intelligence Program**

| | |
|---|---|
| DGSTA Defence Intelligence Organisation | 1 |
| Manager, Information Centre, Defence Intelligence Organisation | 1 |

**Corporate Support Program**

| | |
|---|---|
| Library Manager, DLS-Canberra | Doc Control Sheet |

| | |
|---|---|
| **Universities and Colleges** | |
| Australian Defence Force Academy | 1 |
| Library | 1 |
| Head of Aerospace and Mechanical Engineering | 1 |
| Serials Section (M list)), Deakin University Library | 1 |
| Hargrave Library, Monash University | Doc Control Sheet |
| Librarian, Flinders University | 1 |
| **Other Organisations** | |
| NASA (Canberra) | 1 |
| AusInfo | 1 |
| State Library of South Australia | 1 |
| Parliamentary Library, South Australia | 1 |

**OUTSIDE AUSTRALIA**

| | |
|---|---|
| **Abstracting and Information Organisations** | |
| Library, Chemical Abstracts Reference Service | 1 |
| Engineering Societies Library, US | 1 |
| Materials Information, Cambridge Scientific Abstracts US | 1 |
| Documents Librarian, The Center for Research Libraries, US | 1 |
| **Information Exchange Agreement Partners** | |
| Acquisitions Unit, Science Reference and Information Service, UK | 1 |
| Library - Exchange Desk, National Institute of Standards and Technology, US | 1 |
| SPARES | 5 |
| **Total number of copies:** | **53** |

**DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION**
**DOCUMENT CONTROL DATA**

| | |
|---|---|
| 1. PRIVACY MARKING/CAVEAT (OF DOCUMENT) | N/A |
| 2. TITLE | Component-Based Engineering of Knowledge-Enabled Systems |
| 3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) | Document (U)<br>Title (U)<br>Abstract (U) |
| 4. AUTHOR(S) | D.P.J. Goodburn, T.R Pattison and R.J. Vernik |
| 5. CORPORATE AUTHOR | Electronics and Surveillance Research Laboratory<br>PO Box 1500<br>Salisbury SA 5108 Australia |
| 6a. DSTO NUMBER | DSTO-GD-0271 |
| 6b. AR NUMBER | AR-011-705 |
| 6c. TYPE OF REPORT | General Document |
| 7. DOCUMENT DATE | February 2001 |
| 8. FILE NUMBER | N9505/19/265 |
| 9. TASK NUMBER | JNT 00/130 |
| 10. TASK SPONSOR | DGC4 |
| 11. NO. OF PAGES | 28 |
| 12. NO. OF REFERENCES | 16 |
| 13. URL ON THE WORLDWIDE WEB | http://www.dsto.defence.gov.au/corporate/reports/DSTO-GD-0271.pdf |
| 14. RELEASE AUTHORITY | Chief, Information Technology Division |

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

*Approved for public release*

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, SALISBURY, SA 5108

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS

Knowledge based systems
Information warfare
Intelligent agents
Software engineering
Expert systems

19. ABSTRACT

Australia's Strategic Policy assigns the highest capability development priority to the achievement of the "knowledge edge" over our adversaries and identifies a single Command Support System that exhibits *flexibility*, *cost-effectiveness*, *robustness* and *adaptivity* as being an important enabler of the knowledge edge. This emphasis on the knowledge edge is reinforced in the Defence 2000 White Paper [2], which states that the knowledge edge "will be the foundation of our military capability over the coming decades". The Software Systems Engineering (SSE) group of ITD is currently conducting research into new software engineering principles and practices, collectively referred to as *component-based software engineering* (CBSE), which are expected to meet the requirements of cost-effectiveness and flexibility in the development of command support systems. In addition, *knowledge-based techniques*, and in particular *intelligent agents*, provide the opportunity to incorporate adaptivity and robustness into software systems through the use of machine-learning, automated reasoning and encapsulation of domain knowledge. In this paper we outline an extension of the Software Systems Engineering Group's research into CBSE to investigate ways in which intelligent agents can be used to exploit the combined advantages of component-based software engineering and knowledge-based techniques for the development of software for military applications.